

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re Patent Application of:

Sung-hyu HAN, et al.

Application No.:

Group Art Unit:

Filed: December 9, 2003

Examiner:

For: DEMODULATION APPARATUS AND METHOD USING CODE TABLE THAT  
DECREASES COMPLEXITY

**SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN  
APPLICATION IN ACCORDANCE  
WITH THE REQUIREMENTS OF 37 C.F.R. § 1.55**

Commissioner for Patents  
PO Box 1450  
Alexandria, VA 22313-1450

Sir:

In accordance with the provisions of 37 C.F.R. § 1.55, the applicant(s) submit(s) herewith  
a certified copy of the following foreign application:

Korean Patent Application No(s). 2002-81028

Filed: December 18, 2002

It is respectfully requested that the applicant(s) be given the benefit of the foreign filing  
date(s) as evidenced by the certified papers attached hereto, in accordance with the  
requirements of 35 U.S.C. § 119.

Respectfully submitted,

STAAS & HALSEY LLP

Date: December 9, 2003

By: 

Michael D. Stein  
Registration No. 37,240

1201 New York Ave, N.W., Suite 700  
Washington, D.C. 20005  
Telephone: (202) 434-1500  
Facsimile: (202) 434-1501



별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto is a true copy from the records of the Korean Intellectual Property Office.

출원 번호 : 10-2002-0081028  
Application Number

출원 년 월 일 : 2002년 12월 18일  
Date of Application  
DEC 18, 2002

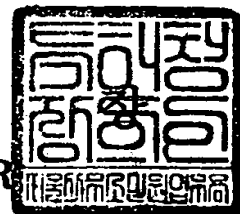
출원인 : 삼성전자주식회사  
Applicant(s) SAMSUNG ELECTRONICS CO., LTD.



2003      07      19  
년      월      일

특      허      청

COMMISSIONER





1020020081028

출력 일자: 2003/7/21

**【서지사항】**

<b>【서류명】</b>	특허출원서
<b>【권리구분】</b>	특허
<b>【수신처】</b>	특허청장
<b>【참조번호】</b>	0003
<b>【제출일자】</b>	2002. 12. 18
<b>【국제특허분류】</b>	G11B
<b>【국제특허분류】</b>	H03M
<b>【발명의 명칭】</b>	복잡도를 줄인 코드 테이블을 사용하는 복조 장치 및 그 방법
<b>【발명의 영문명칭】</b>	Apparatus and method for demodulating using a code table decreased a complexity
<b>【출원인】</b>	
<b>【명칭】</b>	삼성전자 주식회사
<b>【출원인코드】</b>	1-1998-104271-3
<b>【대리인】</b>	
<b>【성명】</b>	이영필
<b>【대리인코드】</b>	9-1998-000334-6
<b>【포괄위임등록번호】</b>	1999-009556-9
<b>【대리인】</b>	
<b>【성명】</b>	이해영
<b>【대리인코드】</b>	9-1999-000227-4
<b>【포괄위임등록번호】</b>	2000-002816-9
<b>【발명자】</b>	
<b>【성명의 국문표기】</b>	한성휴
<b>【성명의 영문표기】</b>	HAN, Sung Hyu
<b>【주민등록번호】</b>	710223-1063223
<b>【우편번호】</b>	135-220
<b>【주소】</b>	서울특별시 강남구 수서동 신동아아파트 704동 1203호
<b>【국적】</b>	KR
<b>【발명자】</b>	
<b>【성명의 국문표기】</b>	김기현
<b>【성명의 영문표기】</b>	KIM, Ki Hyun
<b>【주민등록번호】</b>	691220-1053119

【우편번호】	305-503
【주소】	대전광역시 유성구 송강동 200-1 한솔아파트 104동 603호
【국적】	KR
【발명자】	
【성명의 국문표기】	박인식
【성명의 영문표기】	PARK, In Sik
【주민등록번호】	570925-1093520
【우편번호】	442-470
【주소】	경기도 수원시 팔달구 영통동 신나무실 615동 801호
【국적】	KR
【발명자】	
【성명의 국문표기】	이윤우
【성명의 영문표기】	LEE, Yoon Woo
【주민등록번호】	650525-1177713
【우편번호】	442-470
【주소】	경기도 수원시 팔달구 영통동 현대아파트 726동 303호
【국적】	KR
【취지】	특허법 제42조의 규정에 의하여 위와 같이 출원합니다. 대 리인 이영 필 (인) 대리인 이해영 (인)
【수수료】	
【기본출원료】	18 면 29,000 원
【가산출원료】	0 면 0 원
【우선권주장료】	0 건 0 원
【심사청구료】	0 항 0 원
【합계】	29,000 원
【첨부서류】	1. 요약서·명세서(도면)_1통

**【요약서】****【요약】**

본 발명에는 복잡도를 줄인 코드 테이블을 사용하는 복조 장치 및 그 방법이 개시되어 있다. 본 발명의 복조 장치는 입력되는 신호로부터 코드워드를 구성하는 각각의 비트에 대하여 각 비트의 확률을 나타내는 값을 계산하여 데이터 워드의 소프트한 복조값을 출력하는 소프트 복조부를 포함하고, 소프트 복조부는 코드워드를 엔트리로 하여 엔트리들을 서로 비슷한 것끼리 이웃하여 효율적으로 배치된 코드 테이블을 이용하여 데이터 워드의 소프트한 복조값을 계산함으로써 효율적으로 구성된 코드 테이블을 사용함에 의해 계산량이 줄어들어 계산 시간을 단축할 수 있다.

**【대표도】**

도 2

**【명세서】****【발명의 명칭】**

복잡도를 줄인 코드 테이블을 사용하는 복조 장치 및 그 방법{Apparatus and method for demodulating using a code table decreased a complexity}

**【도면의 간단한 설명】**

도 1은 기존의 소프트 복조부를 설명하기 위한 참고 테이블,

도 2는 본 발명이 적용되는 전체 시스템의 블록도,

도 3은 본 발명에 따른 복잡도를 줄인 코드 테이블의 일 예,

도 4는 본 발명에 따른 복잡도를 줄인 코드 테이블의 다른 예이다.

**【발명의 상세한 설명】****【발명의 목적】****【발명이 속하는 기술분야 및 그 분야의 종래기술】**

- <5> 본 발명은 신호 복조 분야에 관한 것으로, 특히 복호 효율을 높이기 위한 소프트 복조기에 사용되는 코드 테이블의 복잡도를 줄이고 이를 이용하는 복조 장치 및 그 방법에 관한 것이다.
- <6> 일반적으로 데이터의 심볼간 간섭(ISI: Inter Symbol Interference)이 존재하는 환경, 예컨대 고밀도 광 기록매체에서 ISI에 의한 문제를 고려하여 런길이 제한(run-length limited: RLL) 코드를 사용하여 데이터를 변조한다.

- <7> 종래의 RLL 코드를 복호화하는 경우를 보면 채널 복조기, 예컨대 비터비 복호기에서 채널을 통해 입력된 신호로부터 코드워드들을 검출하고, RLL 복호기에서 디코딩 테이블을 이용하여 코드워드들을 데이터 워드들로 복호화한다.
- <8> 최근에는 소프트 복조기 및 소프트 복호기가 사용되고 있다. 상술한 비터비 복호기의 출력은 에러를 포함하는 코드워드 자체가 되어 "1" 또는 "-1"의 값만 가진다. 이러한 방식을 하드 복조라고 할 때 이에 대응하여 소프트 복조 방식에서는 소프트 복조기가 채널 신호를 입력받아 코드워드의 확률값을 출력한다. 즉, "1" 또는 "-1" 이 아닌 "0.8", "-0.8" 등의 아날로그적인 값을 가지며 그 자체에 코드값이 "1" 또는 "-1"이 될 확률이 내포된다. 소프트 복조기는 이러한 코드워드의 확률값을 나타내는 데이터를 입력받아 데이터 워드의 확률값을 출력하고, 터보 복호기와 같은 소프트 복호기에서 데이터 워드의 확률값을 입력받아 최종적으로 데이터 워드로 복호화한다.
- <9> 소프트 변조를 필요로 하는 터보 디코딩에 관한 참조 문헌으로는 Laura L. McPheters and Steven W. McLaughlin, "Turbo-Coded Optical Recording Channels with DVD Minimum Mark Size", IEEE Transactions on Magnetism, vol.38, no. 1, January pp 298-302이 있다.
- <10> 한편, 상술한 소프트 복조기에서는 코드워드의 확률값을 나타내는 데이터를 입력받아 데이터 워드를 구성하는 각각의 비트의 확률을 나타내는 값인 LLR(Log Likelihood Ratio)을 구하는 과정을 도 1에 도시된 테이블을 참조하여 설명하면 다음과 같다.
- <11> 도 1은 코드 레이트가 2/3이고, RLL(1,7) 코드의 디코딩 테이블을 나타내는 도면으로서, 첫 번째 행은 디코딩된 결과인 2비트 길이의 데이터 워드를 나타내고,

나머지 값들은 각각의 데이터 워드에 대응하는 코드워드들을 나타낸 것이다. 2비트의 데이터 워드를 복호화하기 위해서 9비트의 코드워드를 대상으로 복호화하는 예를 보이고 있다.

<12> 먼저, LLR값을 구하기 위해 APP(A Posteriori Probability)( $d_k=1$ )과 APP( $d_k=0$ )을 각각 계산한다.

<13> APP( $d_k=1$ )은 복조된 데이터가 "1"일 확률을 나타내는 값이며, APP( $d_k=0$ )은 복조된 데이터가 "0"일 확률을 나타내는 값이다. 이 값은 다음과 같은 방법으로 구한다. 만일 데이터 워드를 결정하기 위한 코드워드의 길이가 t비트 이라면, 데이터 워드의 1 비트가 1이 되도록 하는 코드워드에 대해 각 비트별로  $(r_m - (2 * c_m - 1))^2$ 의 값을 구하여 더한다. 여기서  $m=0 \dots t-1$ 의 값이다. 이 값들을 상술한 데이터 워드의 1 비트가 1이 되도록 하는 모든 코드워드 M개에 대해 지수(exponential)값을 취하여 더하면 APP( $d_k=1$ )의 값이 구해진다. 이를 식으로 나타내면 아래와 같다.

<14> **【수학식 1】** 
$$APP(d_k=1) = \sum_{j \in S_1(k)} \exp[r'_m - (2 * c'_m - 1)^2]$$

<15> 여기서, j는 j번째 데이터 워드가 1이 된다는 의미이다.  $S_1(k)$ 는 도 1에 도시된 테이블에서  $d_k d_{k+1}=10$ 과  $d_k d_{k+1}=11$ 에 해당하는 엔트리들의 집합이다.

<16> APP( $d_k=0$ )에 대해서도 마찬가지로 방법으로 구하는 데 즉, 아래 수학식 2에 도시된 바와 같다.

<17> **【수학식 2】** 
$$APP(d_k=0) = \sum_{j \in S_0(k)} \exp[r'_m - (2 * c'_m - 1)^2]$$



<18> 여기서,  $S_0(k)$ 는 도 1에 도시된 테이블에서  $d_k d_{k+1}=00$ 과  $d_k d_{k+1}=01$ 에 해당하는 엔트리들의 집합이다.

<19>  $LLR(d_k)$ 는 수신된 코드워드에서 복조될 데이터 워드  $d_k$ 의 1비트가 0인 확률과 1인 확률의 비에 대해 지수를 취한 값이므로 다음과 같이 나타낼 수 있으며, 이 값을 소프트 복조기의 출력으로 사용한다.

<20>

$$LLR(d_k) = \log\left(\frac{\Pr(d_k = 1 | R)}{\Pr(d_k = 0 | R)}\right), R = r_0, r_1, \dots, r_{t-1}$$

$$= \log\left(\frac{\Pr(R | d_k = 1)}{\Pr(R | d_k = 0)}\right)$$

$$\approx \log\left(\frac{\sum_{j \in S_1(k)} \Pr(R | C^j \text{transmitted})}{\sum_{j \in S_0(k)} \Pr(R | C^j \text{transmitted})}\right)$$

【수학식 3】

<21>  $C^j$ 의 발생빈도를 무시하고 동일하게 발생한다고 가정하고, 확률  $\Pr(R | C^j)$ 는 다음 수학식 4와 같이 표현된다.

<22>

$$\Pr(R | C^j) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^t \exp\left[-\frac{1}{2\sigma^2} \sum_{m=0}^{t-1} (r'_m - (2 * c'_m - 1))^2\right]$$

【수학식 4】

<23> 따라서, 수학식 3은 수학식 5와 같이 다시 쓸 수 있다.

<24>

$$\log\left(\frac{\sum_{j \in S_1(k)} \Pr(R | C^j \text{transmitted})}{\sum_{j \in S_0(k)} \Pr(R | C^j \text{transmitted})}\right) = \log\left(\frac{\sum_{j \in S_1(k)} \exp\left[\frac{1}{\sigma^2} \sum_{m=0}^{t-1} r'_m * (2 * c'_m - 1)\right]}{\sum_{j \in S_0(k)} \exp\left[\frac{1}{\sigma^2} \sum_{m=0}^{t-1} r'_m * (2 * c'_m - 1)\right]}\right)$$

$$= \log \sum_{j \in S_1(k)} \exp\left[\frac{1}{\sigma^2} \sum_{m=0}^{t-1} r'_m * (2 * c'_m - 1)\right] - \log \sum_{j \in S_0(k)} \exp\left[\frac{1}{\sigma^2} \sum_{m=0}^{t-1} r'_m * (2 * c'_m - 1)\right]$$

$$\approx \max_{j \in S_1(k)} \left[\frac{1}{\sigma^2} \sum_{m=0}^{t-1} r'_m * (2 * c'_m - 1)\right] - \max_{j \in S_0(k)} \left[\frac{1}{\sigma^2} \sum_{m=0}^{t-1} r'_m * (2 * c'_m - 1)\right]$$

【수학식 5】

<25> 위 수학식 5는 아래 수학식 6에 표현된 최대 지수 근사(Max Log Approximation) 정리를 이용하고,  $1/\sigma^2$  무시해도 결과는 동일한 성능을 나타내므로 수학식 7과 같이 나타낼 수 있다.

&lt;26&gt;

【수학식 6】  $\log(e^{\delta_1} + e^{\delta_2} + \dots + e^{\delta_n}) \approx \max_{j \in \{1, 2, \dots, n\}} \delta_j$

&lt;27&gt;

즉, 지수 계산이 복잡할 때는 APP 계산시 지수 계산을 하지 않고 최대값을 취하고, 수학식 3에서 표현된 제산 연산 대신 APP(d=1)-APP(d=0)의 값을 계산해서 LLR(d<sub>k</sub>)를 구한다. 즉,

&lt;28&gt;

【수학식 7】  $LLR(d_k) = \max_{j \in S_1(k)} [\sum_{m=0}^{L-1} r'_m * 2(c'_m - 1)] - \max_{j \in S_0(k)} [\sum_{m=0}^{L-1} r'_m * 2(c'_m - 1)]$

&lt;29&gt;

위 수학식 7의  $\max_{j \in S_1(k)} [\sum_{m=0}^{L-1} r'_m * 2(c'_m - 1)]$ 에서, S<sub>1</sub>(k)와 같은 집합은 도 1에 도시된 테이블로 나타난다.

&lt;30&gt;

또한, LLR(d<sub>k+1</sub>)은 LLR(d<sub>k</sub>) 수학식 7에서 k대신 k+1을 대입하면 된다. 즉,

&lt;31&gt;

【수학식 8】  $LLR(d_{k+1}) = \max_{j \in S_1(k+1)} [\sum_{m=0}^{L-1} r'_m * 2(c'_m - 1)] - \max_{j \in S_0(k+1)} [\sum_{m=0}^{L-1} r'_m * 2(c'_m - 1)]$

&lt;32&gt;

여기서, S<sub>0</sub>(k+1)은 도 1에 도시된 테이블에서 d<sub>k</sub>d<sub>k+1</sub>=00과 d<sub>k</sub>d<sub>k+1</sub>=10에 해당하는 엔트리들의 집합이고, S<sub>1</sub>(k+1)은 도 1에 도시된 테이블에서 d<sub>k</sub>d<sub>k+1</sub>=01과 d<sub>k</sub>d<sub>k+1</sub>=11에 해당하는 엔트리들의 집합이다.

&lt;33&gt;

상술한 바와 같이, 종래에는 소프트 복조기에 사용되는 APP 디코딩 테이블의 사이즈가 커지면 LLR을 계산하는 데 계산 시간이 늘어나고, 디코딩 테이블의 복잡도가 크게 증가하는 문제점이 있었다.

**【발명이 이루고자 하는 기술적 과제】**

<34> 따라서, 본 발명의 목적은 복호 효율을 높이기 위한 소프트 복조기에 사용되는 코드 테이블의 복잡도를 줄이고, 이를 이용하는 복조 장치 및 그 방법을 제공하는 데 있다.

<35> 본 발명의 다른 목적은 복잡도를 줄인 테이블을 구성해서 런길이제한을 가지는 코드의 소프트 복조 출력을 구하는 복조 장치 및 그 방법을 제공하는 데 있다.

**【발명의 구성 및 작용】**

<36> 본 발명에 따르면, 상기 목적은 제1 소정 비트 길이의 데이터 워드가 제1 소정 비트 길이 이상인 제2 소정 비트 길이의 코드워드로 변조된 코드를 복조하는 장치에 있어서: 입력되는 신호로부터 코드워드를 구성하는 각각의 비트에 대하여 상기 각 비트의 확률을 나타내는 값을 계산하여 데이터 워드의 소프트한 복조값을 출력하는 소프트 복조부를 포함하고, 상기 소프트 복조부는 코드워드를 엔트리로 하여 엔트리들을 서로 비슷한 것끼리 이웃하여 효율적으로 배치된 코드 테이블을 이용하여 상기 데이터 워드의 소프트한 복조값을 계산하는 것을 특징으로 하는 복조 장치에 의해 달성된다.

<37> 본 발명의 다른 분야에 따르면, 상기 목적은 제1 소정 비트 길이의 데이터 워드가 제1 소정 비트 길이 이상인 제2 소정 비트 길이의 코드워드로 변조된 코드를 복조하는 방법에 있어서: 입력되는 신호로부터 코드워드를 구성하는 각각의 비트에 대하여 상기 각 비트의 확률을 나타내는 값을 계산하여 데이터 워드의 소프트한 값을 출력하는 소프트 복조값 출력 단계를 포함하고, 상기 소프트 복조값 출력 단계에서는 코드워드를 엔트리로 하여 엔트리들을 서로 비슷한 것끼리 이웃하여 효율적으로 배치된 코드 테이블을

이용하여 상기 데이터 워드의 소프트 복조값을 계산하는 것을 특징으로 하는 복조 방법에 의해 달성된다.

<38> 이하, 첨부된 도면을 참조하여 본 발명의 바람직한 실시 예를 설명하기로 한다.

<39> 도 2는 본 발명이 적용되는 전체 시스템의 블록도로서, 부호부(10), 인터리버(20), 변조부(30), 전송 채널(40), 소프트 복조부(50), 디인터리버(60), 반복 복호부(70)로 되어 있지만 다양한 변형된 예가 더 있을 수 있으며, 예를 들어 인터리버(20)와 디인터리버(60)가 구성되지 않을 수도 있으며, 전송 채널(40) 후단에 소프트 채널 검출기가 더 구성될 수 있다.

<40> 도 2를 참조하면, 부호부(10)는 입력되는 데이터를 에러 정정을 수행하기 위한 소정의 부호화 방식, 예컨대 LDPC(Low Density Parity Check Code) 부호화 방식 및/또는 터보(Turbo) 부호화 방식과 같은 소프트 부호화 방식을 이용하여 부호화한다. 인터리버(20)는 부호화된 데이터를 소정의 인터리브 순서대로 인터리빙한 후 인터리브된 데이터를 변조부(30)에 제공하고, 변조부(30)는 인터리브된 데이터를 적당한 방법 예컨대 RLL 코드 등을 사용하여 변조한다. 변조된 데이터는 기록 매체 등의 각종 전송 채널(40)을 통과하면서 왜곡된 신호의 형태로 소프트 복조부(50)에 입력된다.

<41> 소프트 복조부(50)는 전송 채널(40)을 통해 입력된 코드워드를 구성하는 각각의 비트에 대하여 상기 비트의 확률을 나타내는 값을 각각의 비트에 대하여 계산하여 데이터 워드의 소프트한 값 그대로 디인터리버(60)에 제공한다. 디인터리버(60)는 소프트 복조 출력값을 인터리빙되기 이전의 원래의 순서로 복원하기 위해 디인터리빙한다. 반복 복호기(70)는 디인터리브된 데이터를 입력하여 소정의 부호화 방식에 대응하여 소프트 복호화를 수행한 후 복호된 데이터와 복호 성공 여부를 나타내는 부가 정보를 출력하고, 이

복호 성공 여부를 나타내는 부가 정보에 따라 반복 복호 방식을 수행하여 신호의 검출 및 에러 정정 능력을 높인다.

<42> 도 3은 도 2에 도시된 소프트 복조부(50)에 사용되는 본 발명에 따른 복잡도를 줄인 코드 테이블의 일 예를 설명하기 위한 도면으로서, 도 3의 (a)는 본 발명의 이해를 돕기 위하여 기존의 디코딩 테이블을 보이고 있고, 도 3의 (b)는 도 3의 (a)에 도시된 코드워드를 재배치하여 복잡도를 줄인 본 발명에 따른 디코딩 테이블로서, 2비트의 데이터 워드를 복호화하기 위해서 16비트의 코드워드를 대상으로 복호화하는 예를 보이고 있다.

<43> 도 3의 (a)에 도시된 바와 같이 기존의 테이블의 코드워드 배치방법은 단순히 엔트리들을 나열하는 것이다. 하지만 테이블에 속한 엔트리들을 서로 비슷한 것 끼리 이웃하여 배치하도록 도 3의 (b)에 도시된 바와 같이 다시 구성하면, 즉, 엔트리들을 칼럼으로 여러 개로 분할하고, 분할된 엔트리들을 공통 부분이 생기도록 배치하고, 이웃하는 공통 부분은 한번만 나타나게 배치한다. 이렇게 구성된 테이블을 이용하여 한 번 계산된 부분은 다시 계산하지 않고 재사용하면 계산량을 줄일 수 있다. 도 3의 (b)에 도시된 테이블에서는 하나의 칼럼을 3등분했지만 2등분 또는 6등분을 할 수도 있다.

<44> 따라서, 도 3의 (a) 및 (b)에 도시된 테이블은 서로 같은 내용을 갖더라도 엔트리들의 효율적인 배치에 의해 LLR을 계산하는 데 계산량이 감소하고 계산 시간이 빨라진다. 예를 들어, 위 수학적 식 7에 도시된  $S_1(k)$ 가 도 3의 (a)에 도시된 기존의 테이블처럼 8개의 원소를 가지는 집합으로 구성되어 있다고 가정해 보자. 도 3의 (a)에 도시된 테이블을 참조하여 LLR을 계산하면, 덧셈 또는 뺄셈이  $8 \times 17 = 136$ 번 필요하다. 그러나, 도 3의 (b)에 도시된 테이블을 사용하면 공통으로 계산되는 부분은 다시 계산하지

않아도 되므로 덧셈 또는 뺄셈의 횟수를 줄일 수 있다. 즉, 0 0 0 0 0 0 에 해당하는 계산은 한번의 계산으로 4번 사용되고, 0 1 0 0 0 0 에 해당하는 계산은 한번의 계산으로 2번 사용되기 때문이다. 이런 식으로 계산을 하면 덧셈 또는 뺄셈이  $14 \times 5 + 12 = 82$ 번 필요하다.

<45> 도 4는 도 2에 도시된 소프트 복조부(50)에서 사용되는 본 발명에 따른 복잡도를 줄인 코드 테이블의 다른 예를 설명하기 위한 도면으로서, 도 3에 도시된 테이블보다는 보다 일반화시키고, 표기된 a,b,c,d,e,f는 각각 3비트를 나타내며, 2비트의 데이터 위드를 복호화하기 위해서 16비트의 코드위드를 대상으로 복호화하는 예를 보이고 있다.

<46> 도 4의 (a)는 본 발명의 이해를 돕기 위하여 기존의 디코딩 테이블을 보이고 있고, 도 4의 (b)는 도 4의 (a)에 도시된 코드위드를 재배치하여 복잡도를 줄인 디코딩 테이블의 예를 보이고 있으며, 도 4의 (b)에서는 하나의 칼럼을 2등분했지만 3등분 또는 6등분을 할 수도 있다.

<47> 따라서, 도 4의 (a)에 도시된 테이블을 참조하여 LLR을 계산하면, 덧셈 또는 뺄셈이  $10 \times 17 = 170$ 번 필요하고, 최대 함수(Max Function)의 계산도 9번 필요하다. 그러나, 효율적으로 배치된 도 4의 (b)에 도시된 본 발명에 따른 테이블을 참조하여 LLR을 계산하면 덧셈 또는 뺄셈이  $10 \times 8 + 2 = 82$ 번 필요하고, 최대 함수의 계산은 7번 필요하다.

#### 【발명의 효과】

<48> 상술한 바와 같이, 본 발명은 채널의 출력 신호 혹은 소프트 채널 검출기의 출력 신호를 입력하여 RLL 코드의 소프트 복조를 행함에 있어서 소프트 복조 결과는 기존의

방법과 동일하지만 기존의 방법과 달리 효율적으로 구성된 디코딩 테이블을 사용함으로써 계산량이 줄어들어 계산 시간을 단축할 수 있는 효과가 있다.

**【특허청구범위】****【청구항 1】**

제1 소정 비트 길이의 데이터 워드가 제1 소정 비트 길이 이상인 제2 소정 비트 길이의 코드워드로 변조된 코드를 복조하는 장치에 있어서:

입력되는 신호로부터 코드워드를 구성하는 각각의 비트에 대하여 상기 각 비트의 확률을 나타내는 값을 계산하여 데이터 워드의 소프트한 복조값을 출력하는 소프트 복조부를 포함하고,

상기 소프트 복조부는 코드워드를 엔트리로 하여 엔트리들을 서로 비슷한 것끼리 이웃하여 효율적으로 배치된 코드 테이블을 이용하여 상기 데이터 워드의 소프트한 복조값을 계산하는 것을 특징으로 하는 복조 장치.

**【청구항 2】**

제1항에 있어서, 상기 코드 테이블의 각 엔트리들에 대해 하나의 칼럼을 소정수로 분할해서 배치하는 것을 특징으로 하는 복조 장치.

**【청구항 3】**

제2항에 있어서, 상기 분할된 엔트리들을 공통 부분이 생기도록 배치하는 것을 특징으로 하는 복조 장치.

**【청구항 4】**

제3항에 있어서, 이웃하는 상기 공통 부분은 한번만 나타나게 하는 것을 특징으로 하는 복조 장치.



**【청구항 5】**

제1항에 있어서, 상기 소프트 복조부는 상기 코드 테이블을 이용하여 소프트한 복조값을 계산할 때 한 번 계산된 부분은 다시 계산하지 않고, 재사용하는 것을 특징으로 하는 복조 장치.

**【청구항 6】**

제1항에 있어서, 상기 입력되는 신호는 전송 채널을 통해 출력되거나 소프트 채널 검출기를 통해 출력되는 런길이제한을 갖는 코드 신호인 것을 특징으로 하는 복조 장치.

**【청구항 7】**

제1 소정 비트 길이의 데이터 워드가 제1 소정 비트 길이 이상인 제2 소정 비트 길이의 코드워드로 변조된 코드를 복조하는 방법에 있어서:

입력되는 신호로부터 코드워드를 구성하는 각각의 비트에 대하여 상기 각 비트의 확률을 나타내는 값을 계산하여 데이터 워드의 소프트한 값을 출력하는 소프트 복조값 출력 단계를 포함하고,

상기 소프트 복조값 출력 단계에서는 코드워드를 엔트리로 하여 엔트리들을 서로 비슷한 것끼리 이웃하여 효율적으로 배치된 코드 테이블을 이용하여 상기 데이터 워드의 소프트 복조값을 계산하는 것을 특징으로 하는 복조 방법.

**【청구항 8】**

제7항에 있어서, 상기 코드 테이블의 각 엔트리들에 대해 하나의 칼럼을 소정수로 분할해서 배치하는 것을 특징으로 하는 복조 방법.

**【청구항 9】**

제8항에 있어서, 상기 분할된 엔트리들을 공통 부분이 생기도록 배치하는 것을 특징으로 하는 복조 방법.

**【청구항 10】**

제9항에 있어서, 이웃하는 상기 공통 부분은 한번만 나타나게 하는 것을 특징으로 하는 복조 방법.

**【청구항 11】**

제7항에 있어서, 상기 소프트 복조값 출력 단계에서는 상기 코드 테이블을 이용하여 소프트 복조값을 계산할 때 한 번 계산된 부분은 다시 계산하지 않고, 재사용하는 것을 특징으로 하는 복조 방법.

**【청구항 12】**

제7항에 있어서, 상기 입력되는 신호는 전송 채널을 통해 출력되거나 소프트 채널 검출 방법을 통해 출력되는 런길이제한 코드 신호인 것을 특징으로 하는 복조 방법.

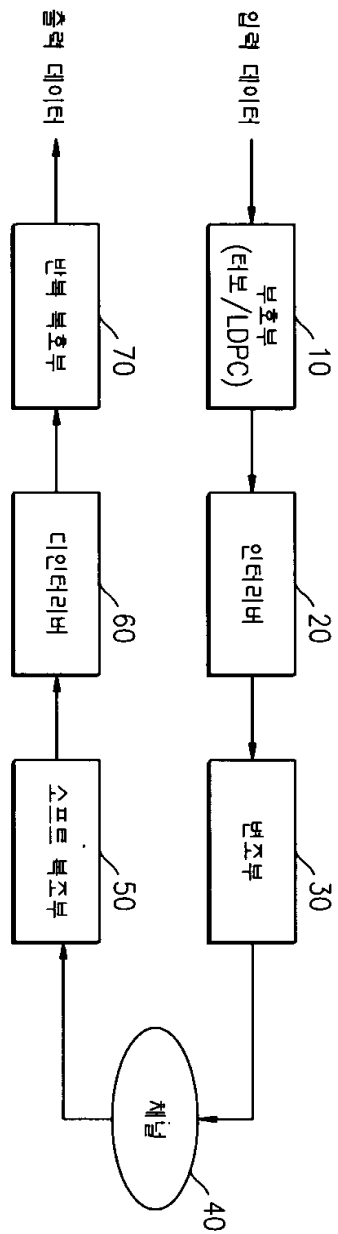
【도면】

【도 1】

	00	01	10	11
1	010010010	100010001	010010001	101010101
...	...	...	...	...
1400	010101001	101010000	001010001	101010010

$S_0(k)$        $S_0(k+1)$        $S_1(k+1)$        $S_1(k)$

【도 2】



【도 3】

(a)

	dkdk+1					
1	0 0 0	0 0 0	0 1 0	0 0 0	1 0 0	0 0 0
2	0 0 1	0 0 1	0 0 0	0 0 0	0 1 0	0 0 0
3	0 0 0	0 0 0	0 1 0	0 0 0	1 0 0	0 0 1
4	0 0 1	0 0 1	0 0 0	0 0 0	0 1 0	0 0 1
5	0 0 0	0 0 0	1 0 0	0 1 0	1 0 0	1 0 1
6	0 0 1	0 0 1	0 1 0	0 0 0	1 0 0	0 0 0
7	0 0 0	0 0 0	1 0 0	0 1 0	1 0 1	0 0 0
8	0 0 1	0 0 1	0 1 0	0 0 0	1 0 0	0 0 1

(b)

	d <sub>k</sub> d <sub>k+1</sub>			
1	0 0 0      0 0 0	0 1 0      0 0 0	1 0 0      0 0 0	
2			1 0 0      0 0 1	
3		1 0 0      0 1 0	1 0 0      1 0 1	
4			1 0 1      0 0 0	
5	0 0 1      0 0 1	0 0 0      0 0 0	0 1 0      0 0 0	
6			0 1 0      0 0 1	
7		0 1 0      0 0 0	1 0 0      0 0 0	
8			1 0 0      0 0 1	

【도 4】

(a)

	dkdk+1
1	a1 b1 c1 d1 e1 f1
2	a1 b1 c1 d2 e2 f2
3	a1 b1 c1 d3 e3 f3
4	a2 b2 c2 d4 e4 f4
5	a2 b2 c2 d5 e5 f5
6	a2 b2 c2 d6 e6 f6
7	a2 b2 c2 d7 e7 f7
8	a3 b3 c3 d1 e1 f1
9	a3 b3 c3 d2 e2 f2
10	a3 b3 c3 d3 e3 f3

(b)

	dkdk+1	
1	a1 b1 c1	d1 e1 f1
	a3 b3 c3	d2 e2 f2
		d3 e3 f3
2	a2 b2 c2	d4 e4 f4
		d5 e5 f5
		d6 e6 f6
		d7 e7 f7